Engineering Contextual Knowledge for Autonomic Pervasive Services

Gabriella Castelli, Marco Mamei,Franco Zambonelli
Dipartimento di Scienze e Metodi dell'Ingegneria
Università di Modena e Reggio Emilia
gabriella.castelli@unimore.it,marco.mamei@unimore.it, franco.zambonelli@unimore.it

Abstract

Services for mobile and pervasive computing should extensively exploit contextual information both to adapt to user needs and to enable autonomic behavior. This raises the problem of how to represent, organize, aggregate, and make available such data to services so as to have it become meaningful and usable knowledge, facilitating the design and development of autonomic pervasive services, and enabling them to acquire high-degrees of context awareness at limited efforts. In this paper, we identify the key software engineering challenges introduced by the need of accessing and exploiting huge amount of heterogeneous contextual information. Following, we survey the relevant proposals in the area of context-aware pervasive computing, data mining and granular computing discussing their potentials and limitations with regard to their adoption in the development of context-aware pervasive services. On these bases, we propose the W4 model for contextual data and show how it can represent a simple yet effective model to enable flexible general-purpose management of contextual knowledge, to facilitate services in achieving high degrees of context-awareness and, overall, to facilitate the design and development of complex pervasive services. A summarizing discussion and the identification of open research directions conclude the paper.

Keywords*: Pervasive Computing, Context-awareness, Autonomic Services, W4 Model, Knowledge Engineering, Critical Survey.*

1. Introduction

Pervasive and mobile computing scenarios consider the possibility of providing users with ubiquitous and on-the-move access to digital services, and of supporting users interactions with their surrounding environments [HenIR06, ManZ06, Cas07]. For this possibility to become practical and satisfying, services should be able to somehow understand situations occurring in the surrounding physical context, autonomously adapt their behavior to the context from which they are requested, and proceed with their execution in an autonomic (i.e., self-organizing, self-adapting, and self-healing) way [ManZ06]. The enforcement of these features necessarily requires both the technology to capture contextual data and the capability of services to exploit such data at the best.

The technology to acquire contextual information is becoming increasingly available, and it will soon become pervasive via the increasing deployment of RFID tags [Wan06], sensor networks [ChoK03, Est02], localization systems [HigB01], users' and organizations' profiles [Phi04]. This fact, together with the increasing success of participatory Web 2.0 tools, will soon make available to services overwhelming amounts of information about facts and events occurring in the physical and social world. Accordingly, the real challenge for future pervasive services is the investigation of principles, algorithms, and tools, via which this growing amount of distributed information can be properly represented, organized, aggregated, and made more meaningful, so as to facilitate the successful exploitation by pervasive services [Bau06].

Future pervasive and autonomic services must evolve from a simple model of “context-awareness”, in which they access isolated pieces of heterogeneous contextual data and are directly in charge of digesting and understanding them, towards a model of “situation-awareness”, in which services access properly structured and organized information reflecting comprehensive contextual knowledge related to a “situation” of interest and which can be exploited in a standardized fashion. The idea is to make available to services a simple yet expressive interface via which to access contextual knowledge, and encapsulate behind such interface all the algorithms and tools to engineer contextual data (i.e., represent, organize, prune and aggregate it). The result is that services can reach, with reduced computational and communication efforts, a comprehensive understanding of “situations” around and, consequently, a higher-degree of adaptability and autonomicity, with the notable software engineering advantage of enforcing a sharp separation of concerns between context exploitation and context management.

In the past few years, a number of different research communities, from pervasive computing, to data mining and granular computing, have somewhat recognized the above problems, and have proposed solutions aimed at engineering large amounts of contextual data and at turning them into usable knowledge [DeyA00, CegR06, DeGM05, Yao06a, Yao06b]. Thus, a first contribution of this paper is to survey such diverse proposals in different areas and critically analyse their potentials and limitations with respect to their exploitation as tools for engineering context information for pervasive services. The result of our analysis is that, despite the potentials of specific approaches for specific problems, none of them can qualify as a fully-fledged general-purpose solution for the challenges raised by pervasive autonomic services.

Following, the second contribution of this paper is to present a novel data model to represent context information, and discuss its potential to act as a general-purpose model to handle several kinds of context information. The model, which we call “W4”, is based on the consideration that most information about the world (i.e., about facts occurring in the world) can be simply represented in terms of four “*W*”s – *Who, What, Where, When*. Despite its simplicity, such a representation enables for very expressive and flexible data usages. In particular, W4 data can be easily queried and accessed by services, tolerate the effective execution of semantic data organization and data aggregation, and can be effectively used to represent both primitive data and high-level knowledge related to a situation.

The remainder of this paper is organized as follows. Section 2 sketches the general scenario we envision for next generation pervasive services, analyzes the issue of context-awareness, and identifies the key challenges for engineering services via suitable models to represent context information. Section 3 surveys the most relevant proposals that, in different areas, are currently providing solutions for context representation and knowledge organization, and analyzes their potentials and limitations. Section 4 introduces the W4 model, and discusses how it can be exploited as a general tool for pervasive services. Section 5 concludes the paper by summarizing the discussion, presenting our current implementation experience with the W4 model, and outlining open research directions.

2. Autonomic Pervasive Services and Context-awareness

Our everyday environments (houses, offices, and cities) are increasingly being populated by a variety of communication-enabled computing devices, forming the basis of a truly pervasive network and generating increasing amounts of information about the physical world and its processes. Embedded sensors and wireless sensor networks [Est02, ChoK03] collect and make available information about physical phenomena, RFID tags can be attached to objects to describe them and to track their usage [Phi04, Wan06]. We, as humans, typically carry on a mobile phone and/or a PDA, possibly a GPS device and some additional wearable sensors, and can generate a lot of information about ourselves and about our own activities and movements [HigB01, War06]. Similar considerations increasingly apply to cars and home furniture. In addition, the success of participatory Web tools (aka Web 2.0 technologies) and of geographic Web tools (e.g., Google Earth and alike), is increasingly making available nearly-up-to-date information about various facts and events occurring in the physical and social worlds.

The above trend paves the way for the design and development of a wide variety of innovative, autonomic and context-aware, pervasive services. However, it also raises peculiar challenges in engineering such services, mostly due to the issue of engineering all available data and turning it into usable knowledge.

2.1. Context-awareness vs. Situation-Awareness

While mobile and ubiquitous computing generally considers the possibility for users to access general digital services from everywhere and on the move, pervasive computing additionally considers exploiting pervasive networks, distributed sensing and possibly actuating infrastructures for the provisioning of services enabling and supporting our interactions with the environment. These include users-level services for on-line monitoring of surrounding world and interacting with it (i.e., "browsing the world" activities [Cas07]), as well as services for enhancing our social experiences in an environment by enabling novel models of localized social interactions [Pen05]. Dually, it also includes infrastructural services embedded in the environment having the goal of making the environment itself capable of perceiving users' activities within and of adapting itself to the users' needs (i.e., ambient intelligence services).

Whatever the case, it is clear that pervasive services have to be inherently context-aware. In fact, collecting information about situations around and acting accordingly is the very core of the activities of pervasive services [ManZ06].

The need for context-awareness also arises when one wants to enforce autonomic behavior in services, i.e., their capability of self-organizing their activities, self-reconfigure and self-heal on need. Given the intrinsic dynamics and decentralization of pervasive scenarios, where components and devices belonging to different stakeholders can come and go at any time and where the structure of the network is inherently dynamic and unreliable, autonomic behavior is necessary to ensure services' continuity without forcing costly and hard to be managed human intervention. However, for such autonomic features to be enabled, services require the capability of understanding what's happening around and react accordingly.

Given that, as stated at the beginning of this section, a number of technologies exists that contribute producing large amounts of contextual information, one may think that achieving context-awareness is simply a non issue. Whatever the data source producing some raw item of contextual information (i.e., "data atom"), all of them contribute populating a large cloud of data atoms and at making it available to services (see Figure 1). A service in need of understanding what's happening around can access (i.e., internalize) the needed data atoms and analyze them to understand what is the current situation of its context.

Unfortunately, such description is far too simplistic and does not emphasize a number of complexities inherent in it. First, the process of data internalization can lead to high communication and computational costs for a service, in that it may require accessing large amounts of data atoms possibly distributed across different devices and analyze which data may serve its current purposes. Second, the process of analyzing retrieved contextual data atoms and turning them into useful knowledge may be non-trivial. In other words, getting access to context information does not automatically imply the capability of reaching "situation-awareness", i.e., the capability of recognizing a situation. With reference to one of the most assessed definitions of context [DeyA00] "context is any information that can be used to characterize the situation of an entity", our claim here is that technological advances are creating a notable gap between "context is any information" and "that can be used to characterize the situation of an entity". That is, acquiring contextual information does not imply the capability of understanding situations, especially in the presence of an overwhelming amount of unrelated contextual data atoms. Such problem is even exacerbated by the increasing heterogeneity of devices and tools that contribute producing contextual information, and by the consequent need of handling heterogeneity in data representation and semantics.

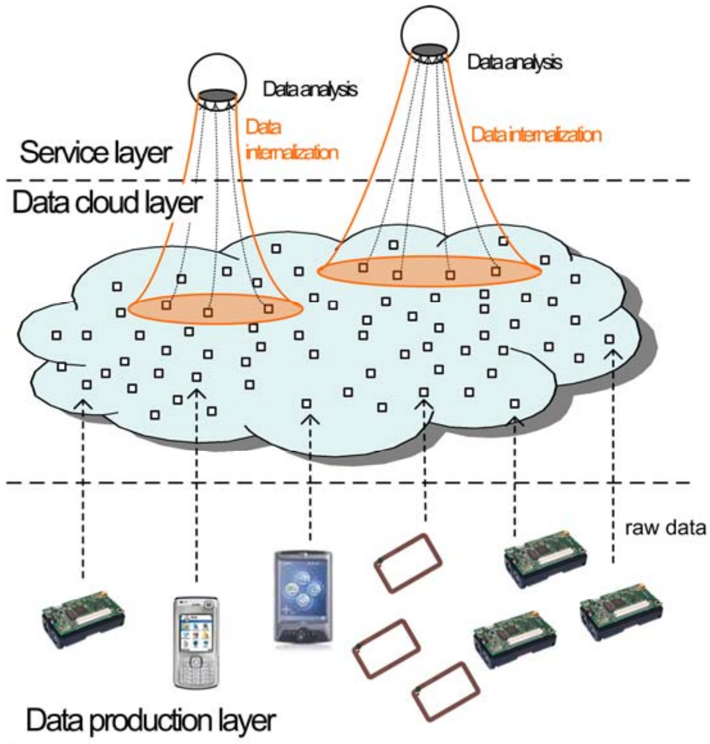

Figure 1: Pervasive devices and sensors make available to services a sort of "data cloud layer", fed with large amounts of heterogeneous data atoms. To serve their purposes, a service needs to retrieve contextual information (i.e., internalize data atoms from the cloud), analyze it to properly understand situations, and finally exploit such knowledge as needed for their own goals.

In our view, for autonomic pervasive services to successfully hit the street, there must be an evolution from a model of simple context-awareness, in which services access isolated pieces of contextual data and are directly in charge of digesting them, towards a model of situation-awareness, in which services access properly structured and organized information, reflecting comprehensive knowledge that is related to a "situation" of interest and which can be exploited in a standardized fashion [Bau06]. With reference to Figure 2, we envision that the access by services to contextual information does no longer occur directly, but rather via a "knowledge network" layer. Such layer should encapsulate mechanisms and tools to analyze and (self-)organize contextual information into sorts of structured collections of related data items, i.e. knowledge networks. Such knowledge networks, by pre-digesting contextual information and by providing compact and expressive information to services, may support them in reaching, with reduced efforts, a comprehensive understanding of "situations" around and, consequently, a higher-degree of adaptability and autonomicity.

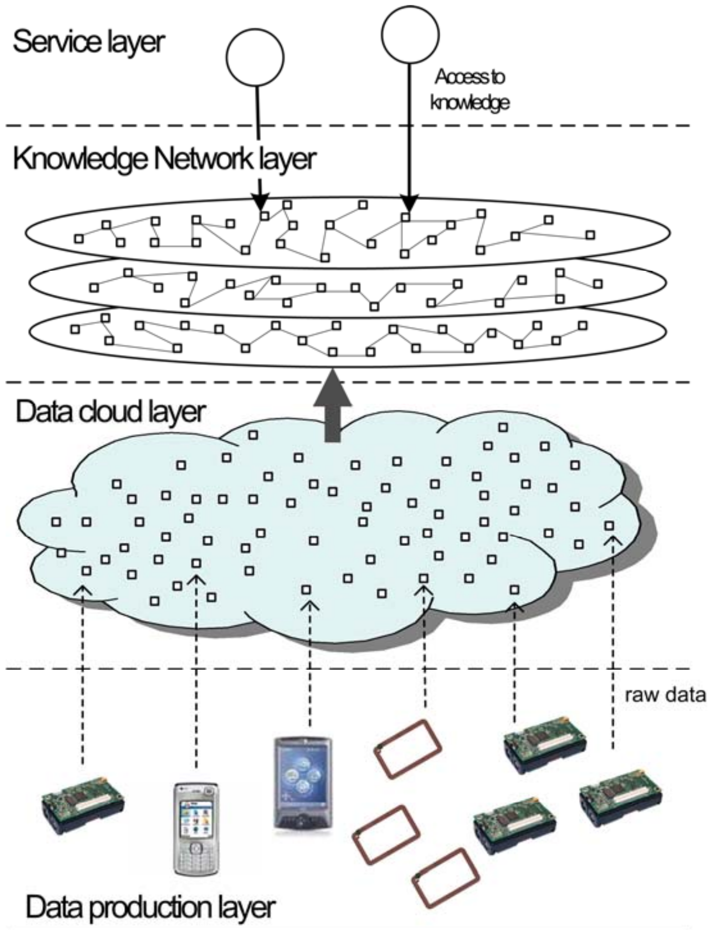

Figure 2: By exploiting a knowledge network layer, services are no longer forced to access the raw data cloud layer. Knowledge organization and analysis is externalized in the middleware, and services are given access to pre-digested information, with a notable complexity reduction.

To make some examples, our idea of knowledge networks includes, among the others, the following possibilities: *(i)* aggregating data atoms to produce compact and possibly multi-level representation of a phenomenon (e.g., averaging the temperature values of a set of sensors in a building and make available compact information about temperatures in single rooms, wings, or in the building as a whole); *(ii)* pruning redundant and/or inconsistent data atoms (e.g., in the presence of a multitude of diverse localization information, from GPS to WiFi based ones, make available to services only the most reliable and accurate ones); *(iii)* relate individual data atoms with each other to increase their informative value (e.g., merging the data read from an RFID-tagged object with those available on some database and with localization information, to produce an accurate description of an object and of its positioning); *(iv)* relate individual data atoms to produce more expressive information (e.g., relating GPS location with data available from the Web to extract information about what the actual activity of a user, such as "Gabriella is at the yearly food exhibition in Modena); *(v)* more in general, perform various kinds of ontology-based context-reasoning activities to extract new knowledge from available data atoms.

From the software engineering viewpoint, an approach based on knowledge networks has the advantage of providing a clear separation of concerns between data analysis and data exploitation. While data analysis and organization is delegated to the knowledge network layer, services are left with the only

duty of exploiting such data to reach specific functionalities. As always in software engineering, this separation of concerns can notably reduce the complexity of developing and maintaining services. A possible criticism of the approach is that it does not eradicate the problem of analyzing large amounts of information, but simply passes it to a different component that either exists at application or at knowledge network level. Although this may be true to some extent, one should consider that, in a distributed setting, knowledge networks can take care of knowledge management duties that would have been otherwise replicated inside each service, with an overall saving in computation and communication efforts.

2.2. Engineering Challenges

For services to effectively achieve situation-awareness, and for our general idea of knowledge networks to become a practical tool for the design and development of autonomic pervasive services, several engineering challenges have to be faced.

Data Model. Our idea of knowledge networks requires the identification of a simple, general-purpose, and uniform model to represent contextual information, i.e., individual data atoms as well as their aggregates. The model should enable representing very diverse facts about the context, typically generated by a variety of heterogeneous sources and at different levels of granularity, in a uniform way. Also, the model should enable ease of manipulation of the data atoms, both by the algorithms and that will be embedded in knowledge networks and by services. Furthermore, the model should enable to deal with incomplete information and information of limited accuracy, since one cannot always assume that all the information within a data atom will be complete and reliable.

Access to Data. The very goal of knowledge networks is to provide knowledge to services and to digest data from any possible contextual data source. First, it is necessary to identify suitable methods (i.e., APIs) by which services can be given access to the knowledge network layer and the information within. Such methods should enable flexible querying of the knowledge network layer, yet should limit the amount of information required by services to fruitfully access it (i.e., services should not rely on a priori information about what data is in the knowledge network layer). Also, given the intrinsic distributed and decentralized nature of contextual information, access to knowledge networks by service should abstract from the actual distribution and allocation of data and knowledge. Second, a similar general API should be provided for enabling data sources (which may include services themselves) to inject new data for feeding the knowledge network layer.

General Approaches for Data Aggregation and Networking. Our idea of knowledge networks is to have it as a "live" layer, which is in charge of continuously and autonomously analyzing information to aggregate data atoms, relate existing knowledge atoms with each other, and extract meaningful knowledge from the available data (i.e., to turn raw "data atoms" into expressive "knowledge atoms" representing a situation from a high level perspective). Thus, it will be necessary to identify algorithmic approaches for performing such analysis without explicit human intervention, i.e.., in a self-organizing and self-configuring way. Also, it is necessary for such algorithmic approaches to be as general as possible, so as to be flexibly adaptable to a variety of application needs without having to re-think from scratch the knowledge network architecture for any new application needs.

Application-specific Views. Strictly related to the above, we have clear in mind that the idea of a single knowledge networks capable, with a limited set of data analysis and aggregation algorithms, of capturing of the possible needs of all possible pervasive services, is illusionary. In a given pervasive environment, a variety of services by different stakeholders may exist and new ones can be deployed at any moment, each with its own goals. Such services may, of course, have very diverse needs for what relate to accessing contextual knowledge, and may require organization/aggregation along different dimensions and based on different algorithms. For instance, some services may be more interested in organizing/analyzing knowledge along the spatial dimension (e.g., for detecting spatial patterns of activities in an environment), some along the temporal dimension (e.g., for detecting temporal patterns of

activities by users), some along some mixed spatio-temporal dimension or along any other application-specific dimension. Thus, and even if a general algorithmic approach has been identified and if some general-purpose algorithmic activity could be identified that is useful for all applications, specific services may require the dynamic instantiation within the knowledge networks of application-specific algorithms for knowledge analysis. Then, a general approach to knowledge networks should account for this possibility and enable the dynamic instantiation of any new knowledge analysis algorithm.

3. Critical Survey of Relevant Related Works

In this section, we survey and analyze several approaches that, in different areas, are somewhat contributing pieces towards the realization of our knowledge network vision. Yet, none of the approaches we have analyzed properly addresses all the identified challenges.

3.1. Context-awareness

In the past few years, a lot of research work in the area of pervasive and ubiquitous computing has focused on identifying proper models for context-aware information, mainly with the goals of engineering usable context representation from low-level sensor data.

Early work in this area, as from Schmidt et al. [Sch99] and Dey et al. [DeyAS01], concentrates on the issue of acquiring context data from sensors and of the processing such data to make it available to processes/services in the form of abstract components. Such approaches thus partially address the data model challenge, in that they recognize the need for a data model able to abstract raw contextual data at higher level. However, they generally miss in identifying a uniform model and a common semantics to describe the data. This forces developers to build new query languages and new components in dependence of the kind of information at hand. In other words, such models do not provide a common semantic to deal with multiple context information in a coherent way, resulting in a complex data model and requiring a great effort to services that has to deal with diverse context representation depending on the supported context sources. The lack of common and general data model consequently implies the lack of a common general-purpose API to access and query data, as well as the lack of any support for data manipulation and organization in multi-level and/or application-specific views.

A different thread of researchers focuses more on the issue of providing rich data models for contextual information and of facilitating querying by services. Schilist et al. [SchAW94] propose a context model in which each contextual data atom is represented by a set of environments variables – each one dealing with a different aspect of the context – that can be accessed in a flexible way. Similarly, Henricksen et al. [HenIR06] model contextual data by making possible to enrich it with several meta-information such as temporal aspects, information imperfection, etc. Such approaches well address the data model challenge, in that they aim at developing a general and flexible data model. However, working with data atoms made up of a long list of elements/variables does not go toward simplicity and generality, which we instead feel should both be goals. Nevertheless, these proposals tend to identify flexible and simple APIs for accessing and querying data, which is fundamental for services to interact with any knowledge/data layer. The issue of relating contextual data atoms with each other and of providing different views to different applications is not generally addressed. More recently, other proposals have adopted a similar endeavor but have considered the issue of adopting specific ontologies to model context-information and enable – other than efficient querying – also efficient context-reasoning [Chen04, GuPZ05], possibly also based upon information, such as location, of specific interest to pervasive computing [Rou06, LeeM07]. Although such approaches tend to be too application-specific, they attribute the importance of linking independent atoms of contextual information (with ontological relations) and of reasoning not only on individual data items but also on their relations, an idea which is fully shared by our knowledge network vision.

An increasing number of research work get inspiration from tuple space data models [AhuCG86] and proposes representing contextual information in the form of tuples, storing them across a set of distributed tuple spaces holding local contextual information, and accessing them via associative (i.e., pattern-matching based) query operations on tuple spaces. Egospaces [JulR06] adopts this perspective, without committing to a specific pre-defined structure for context tuple, which can make it difficult for services to uniformly deal with tuples representing different aspects of the context represented in different formats. However, Egospaces proposes a so called “egocentric” notion of context, in which different services can perceive a different context-dependent representation of the contextual information, depending on their current location. We consider such a feature very important in that it allows to tailor information to specific users’ needs and viewpoints. The Context Fabric model [Hon02] does better than Egospaces as far as tuple structuring is concerned, in that it relies on well-structured context tuples each describing a single piece of context data in terms of entities (people, place, thing), attributes (e.g., the name). Moreover, even if it does not propose solutions for enforcing application-specific views, it considers the possibility of identifying relationships between context tuples. Recent proposals focusing on sensor networks, suggest exploiting a tuple-based approach to flexibly access sensorial information [NewW04, MotP06]. Although not focusing on specific tuple structures, such proposals are of interest in that they consider the possibility of providing application-specific views on sensorial data. The idea is to have services dynamically inject code into the sensors for aggregating/elaborating data within the sensor network, and eventually enabling services to directly access aggregated data according to their own specific needs. In general we consider tuple-based approaches very suitable for organizing and accessing contextual information, but we also think that there is need of more structuring and flexibility than those exhibited by the existing approaches.

Some recent proposals focuses on providing models for contextual data that adopt a uniform well-defined structure, capturing those specific aspects which are of interest by pervasive services, a characteristic which we consider very promising. The approach proposed in the Nexus platform [Leh04] proposes representing different contextual data uniformly accounting for fields such as spatial references and temporal references, for enabling general spatial and temporal queries over a context database. The proposal described in [XuC05] suggests adopting a seven-field data structure to describe the context. The suggested fields include subject, predicate, object, time, area, certainty, freshness, which overall provide quite a complete characterization of contextual information. The system described in [Bra06] proposes describing contextual information contained in RFID tags and read by services in terms of identity, purpose of tag, location of tag, time of information production. Although the system is a special-purpose one, having been applied to RFID tags only, we consider it interesting in that is consider a simple enough yet quite informative structured data model, able to represent in a uniform way different data coming from different sources. Indeed, our proposal accounts for a very similar structuring for contextual information, and enriches it further with a well defined API, and with the possibility of linking data atoms and of providing application-specific views to services.

3.2. Data Mining & Pattern Discovery

As stated in Subsection 2.1, the potentially overwhelming amount of data that can be generated by pervasive sensing infrastructures does not constitute knowledge *per se*, in that services have may have to face complex analysis tasks to get a meaning out of it. Such analysis task, which we think should be delegated to a knowledge network layer, is in the end a sort of data mining process [GalS06].

Data mining concerns analyzing large amounts of typically unrelated data to infer hidden linkages, correlations, rules, constraints, (i.e., broadly speaking, patterns) in such data [Fay96, Bar03], to present such inferences to user for subsequent interpretation, i.e., to have the user give a meaning to data by analyzing the identified patterns. In general, all the mechanisms proposed in this field can be naturally be

employed within the knowledge network layer to extract knowledge from raw data collected by sensing devices.

In the wide data mining research area, a variety of algorithms and approaches have been proposed, most of which rely on a general two step process: identifying relevant sets of related data item within the global dataset and, following, inferring patterns from this sets. This two-steps process can be of inspiration for knowledge networks, and in particular with regard to the need of identifying flexible and general algorithmic approaches for continuously and autonomously aggregate and analyze data atoms. In fact, it may lead to a modular algorithmic approach, in which the two issues of relating data atoms and of extracting higher-level knowledge form such relations can be clearly separated.

Data mining activities may, in general, identify thousands of patterns from data sets, all of them of general interest. With this regard, several researches involve the specialization of association mining fundamentals to address the problems of specific application domains, e.g. spatial or temporal association rules, to limit the number of mined association to the most relevant ones [AlCG07, VerC06]. Of course, for applying the lessons of data mining to pervasive services and knowledge networks, a similar application-specific approach must be taken.

Recently many researchers applied data mining techniques to wireless sensor networks. Sensor network offers new challenges to classical data mining. The large amount of sensed data has to be modelled has a stream, there may be a large number of nodes in the network, calling for decentralized approaches and should account for data losses, and finally the power consumption issue must be considered. Some approaches[BonB05, McS05] focuses on mining sensed data for prediction purpose. [BonB05] proposes a framework for data mining upon sensor network for supervised learning (prediction, classification, etc.) according to a specified level of precision and quality. The framework is based on a two step process: the first step performs aggregation of sensors in clusters, in the second one, each cluster sends the aggregate to a data mining server that performs the analysis. Similarly, [McS05] proposes a framework for prediction based on the flow of local predictors through the network. At the root, predictors are combined via a voting mechanism. Other approaches [GanEH04, KulD05] focuses on the general problem of identification of pattern by using neural networks algorithm in a distributed setting. In particular, these approaches: *(i)* uphold the need for data mining for analyzing the vast amount of data in pervasive computing application, *(ii)* show that decentralized approaches are effective and operable in distributed network with several nodes.

Another trend of research in applying data mining to pervasive computing scenarios consists in analyzing data coming from wearable sensors to infer and predict user's behavior and social interactions.

The work presented in [GiP06] applies data mining techniques to automatically identify social structures among a group of people. In this work, each user is provided with an IR badge continuously transmitting the user ID to other badges nearby. All ID exchanges are recorded, and a data mining algorithm selects those people that tends to stay together over multiple encounters. Such persons are marked as friends or colleagues and thus an inherent social bond between them is automatically detected. Similar works adopt the same principles exploiting different kind of sensors. For example, [Pen05] proposes the use of microphones and IR badges to measure who is talking with whom, and derive social networks and other context information by mining such information. On similar lines, the work on "familiar strangers" [PauG04] records and mine Bluetooth-encounters to identify those people and places that are familiar to us, although we do not know or remember them (e.g., people who usually take the bus with us – lots of Bluetooth encounters, or places in which a lot of people we encounter frequently are located).The work presented in [Pat04] uses data mining techniques (i.e., Bayesian networks) to learn and infer transportation routines of a user provided with a GPS enriched PDA. A data mining engine running on a server, is constantly fed with the users' GPS traces. The server can then infer where the user is probably heading, which bus he has to ride and where he has to get off. The results are sent back to the PDA that can alert the user on missing stops, unusual deviation from the usual track, etc. The systems presented in [Pat05] uses data mining to infer user activities on the basis of the objects the user touches (as revealed by

the sensing of RFID tags stick to the objects). Specifically, the system correlates objects such as "tea-pot", "cup" and "tea-bags", to high-level activities like "preparing tea". Although these approaches are very special-purpose, they again confirm that data mining can be applied to pervasive computing.

As an additional note, typical data mining approaches are human-centered and query-based, i.e., assuming humans are the end users of data mining activities. In pervasive computing, instead, we require automatic methodologies for discovering relations between contextual data and for making these available to computational services. Moreover, conventional data mining assumes the independence between the attributes and the independence between the values of these attributes, but in context-awareness diverse attributes and their corresponding values are often related. Some researches has been done in areas where correlation between attributes exist [GalS06], however they result in human-centered and visual data mining methodologies that are not suitable for pervasive services.

3.3. Data Aggregation Granular Computing

Granular computing is an emerging inter-disciplinary research area that considers the general issue of processing "information granules", i.e., collections of data atoms, and extract knowledge from them. The idea is to organize information granules together based on their similarity, functional properties, spatial/temporal adjacency, or identified regularities in data [Ped01], and eventually provide higher knowledge-level views, at different scales, of the phenomena underlying information granules. Although the strict relations between data mining and granular computing are evident, the latter adopts a more theoretical and inter-disciplinary viewpoint, and specifically focuses on the idea that, at different level of observation and analysis, the same data can provide different knowledge [Yao06a, Yao06b].

The ideas and principles of granular computing have been investigated in many research fields from computer science to psychology: computational intelligence, artificial intelligence, the theory of hierarchy, divide and conqueror, the theory of small groups, etc. Such a wide range of researches demonstrates the potential of granular computing approach. Indeed, the underlying assumption of granular computing is that the basic principles and methodologies are independent from specific problem domains. Three main perspectives motivates the holistic view of granular computing [Yao06a]: *(i)* a philosophical perspective: granular computing as a way of structured thinking models the reality in a multi-level fashion that reflects the granularity structure of the physical world; *(ii)* a practical perspective: granular computing as a method of structured problem solving promotes general and reusable approaches and methodologies in the process of problem solving; *(iii)* a computational perspective: granular computing as a paradigm of information processing leads to more effective computational models.

As theoretical as it can be, granular computing naturally fits also our perspective on context-awareness and situation-awareness, and specifically our idea of knowledge networks. First, the goal of extracting knowledge from information granules directly maps into the idea of introducing a layer above the cloud of contextual data to access higher-level information. Second, the idea of relating information granules based on different characteristics and rules directly maps into our idea of knowledge networks. Finally, the idea of granular computing of providing multi-level views for serving different purposes directly maps into our goal of providing application-specific views of knowledge networks.

To the best of our knowledge, there are no studies directly related to applying granular computing ideas to support context-awareness by pervasive services. Nevertheless, there are studies related to applying granular computing techniques to model spatial and temporal data at different levels of granularity, an issue which is of specific relevance to pervasive services (which are inherently situated in space and time). Camossi et. al. [Cam03] proposes a spatiotemporal data model relying on an extended ODMG model, which provides a uniform management of both moving entities and temporal maps. The model allows for the multi-level management of such data, and also deals with temporal indeterminacy and spatial inaccuracy. Granular GeoGraph [DeGM05] provides a conceptual spatial data model with two granularity dimensions: a purely spatial one and a semantic one. Spatial granularity refers the possible variations of the geometry of an object with respect to different scales, semantic granularities refers to the

possible variations of set of domain objects with respect to the levels of detail requested by different users/applications. It is important to notice that refinements in semantic granularity are not necessarily paired with refinements in spatial resolution.

Although the above researches help us providing some inspiration with spatial and temporal data, context-awareness and situation-awareness especially, involves more contextual factors and more rich set of relations to be taken into account, something which we indeed try to account in our proposal.

4. The W4 Approach

The result of the previous survey is that, despite diverse approaches address specific engineering challenges, none of them propose fully-fledged solutions for the need of modern autonomic pervasive services.

Our proposal for a novel, simple yet effective, data model for expressing contextual knowledge about the world starts from the consideration that any elementary data atoms as well as any higher-level piece of contextual knowledge, in the end, represents a “fact” which has occurred. Accordingly, our proposal simply account that any of such facts – and therefore any data/knowledge atom – can be expressed by means of a simple yet expressive 4-fields tuples (Who, What, Where, When): “someone or something (Who) does/did some activity (What) in a certain place (Where) at a specific time (When)”.

W4 knowledge atoms may be created by proper software agents associated to data sources or sensor. Their four-fields structure is flexible and general enough to uniformly deal with information coming from sources as diverse as embedded devices, cameras, users, or Web 2.0 sites, and can account for adaptation to context and incomplete information (i.e., some of the four fields being unspecified). W4 knowledge atoms, as tuples in tuple spaces, can be stored in suitable shared data spaces, whatever distributed and implemented. Users and services, from everywhere, can retrieve knowledge atoms via a simple API, based on “à la Linda” [AhuCG86] pattern-matching query mechanisms. Such API supports context-aware queries and incomplete information, to enable services to interact with the world and to enforce autonomic and context-aware functionalities. In addition, the simple W4 structure support general distributed algorithms for data aggregation and manipulation, and facilitates the building of semantic knowledge networks and of multiple, application-specific views.

4.1. Data Representation

The four-fields (Who, What, Where, When) of the W4 data model each describes a different aspect of a contextual fact.

The Who field associates a subject to a fact, and may represent a human person (e.g., a username) or an unanimated part of the context acting as a data source (e.g., the ID of an RFID tag). The Who field is represented by a type-value pair, in the form of a string, with an associated namespace that defines the “type” of the entity that is represented. For example, valid entries for this field are: “person:Gabriella”, “tag:tag#567”.

The What field describes the activity performed by the subject. This information can either come directly from the data source (e.g., a sensor is reading a temperature value), or be inferred from other context parameters (e.g., an accelerometer on a PDA can reveal that the user is running), or it can be explicitly supplied by the user. This field is represented as a string containing a predicate:complement statement. For example, valid entries for the What field are: “read:book”, “work:pervasive computing group”, “read:temperature=23”.

The Where field associates a location to the fact. In our model the location may be a physical point represented by its coordinates (longitude, latitude), a geographic region (we currently adopt the PostGIS language to describe such regions), or it can also be a logical place. In addition, context-dependent spatial expressions like “here” or “within:300m” can be used for context-aware querying, as described in the following of this section.

The When field associates a time or a time range to a fact. This may be an exact time/time range (e.g., "2006/07/19:09.00am - 2006/07/19:10.00am"), or a concise description (e.g., 9:28am). For example 9:28am = 2006/07/19:9:28am ± 5min. Also in this case, context-dependent expressions can be defined (e.g., "now", "today", "yesterday", "before") and can be used for context-dependent querying.

The way it structures and organizes information makes the W4 data model general enough to represent data coming from very heterogeneous sources and simple enough to promote ease of management and processing.

4.2. Data Access and Service Engineering

As already stated, it is fundamental to define a simple API for services to access to contextual knowledge and enabling data sources and services to inject new data in the knowledge network layer.

Since knowledge atoms are stored in the form of W4 tuples in a shared data space (or in multiple data spaces), we took inspiration from tuple-space approaches [1] to define the following API:

```
void inject(KnowledgeAtom a);
KnowledgeAtom[] read(KnowledgeAtom a);
```

The inject operation is equivalent to a tuple space "out" operation: an agent accesses the shared data space to store a W4 tuple there.

The read operation is used to retrieve tuples from the data space via querying. A query is represented in its turn as a W4 tuple with some unspecified or only partly specified values (i.e., a template tuple). Upon invocation, the read operation triggers a pattern matching procedure between the template and the W4 tuples that already populate the data space. A vector of all matching tuples – i.e., those for which all the defined fields match those provided in the template – is returned as the result of the query. In any case, pattern matching operations work rather differently from the traditional tuple space model. In fact, our proposal can rely on the W4 structure to enforce expressive context-aware pattern matching operations, which may exploit differentiated mechanisms for the various W4 fields. Current mechanisms work as follows:

- Who and What. Pattern-matching operations in these fields are based on string-based regular expressions. For example, "user:*" will match any user.
- Where. Pattern matching in this field involves spatial operations inspired by PostGIS operations. Basically, the template defines a bounding box (e.g., "circle, center(lonY,latX), radius:500m") and everything within the bounding box matches the template. All tuples with a Where field within the circle will match this field of the template. Contextual places such as "within:300m" can be specified in the template and are translated into actual spatial regions – based on the current location from where the query is performed – before going through the pattern matching.
- When. In this case, the template defines a time interval. Everything that happened within that interval matches the template. Concise time descriptions as well as contextual ones (e.g., "now" or "before") are converted into actual time intervals before pattern matching.

Two simple examples follow to illustrate the querying process.

Let us assume Gabriella is walking in the campus and wants to know if some colleagues are near. She will ask (via a read operation):

```
Who: user:*
What: works:pervasive computing group
Where: circle,center(lonY,latX),radius:500m
When: now
```

Then, she will get in return the tuples representing all the colleagues of her group currently around (at least, of all those colleagues having decided to expose themselves via a W4 tuple).

Similarly, Gabriella can ask if some of her colleagues have gone to work in the morning:

Who: user:* What: works:pervasive computing group Where: office When: 2006/07/19:09am-2006/07/19:10am

We emphasize that the returned answers have not to be "complete" W4 tuples. The pattern matching mechanism also allows for matches between incomplete information. Thus, unlike in traditional tuple space approaches, applications are based on components entering complete and incomplete context information and getting in response refined (but possibly still incomplete) information.

4.3. Data Generation

In the W4 model, we rely on the reasonable assumption that software drivers (or, more in general, software agents) are associated with data sources and are in charge of creating W4 tuples and inserting them in some sorts of shared data spaces. In the end, any data source must be somehow associated with some software to gather and store data items, W4 agents have the additional goal of collecting all the necessary information to produce a W4 tuple which is as accurate and complete as possible. This occurs by sensing and inferring information from all the devices and sources available (e.g., RFID tags, GPS devices, Web services), and by combining them in a W4 tuple. Three simple examples may clarify this concept.

Let us assume Gabriella is walking in the campus park. Agents running on her GPS-equipped PDA, can periodically create the following tuple:

Who: user:Gabriella What: walk:4km/h Where: lonY, latX When: 2006/10/17:10.59am

Where the Who is entered implicitly by the user at the login, What and Where can be derived by the GPS (e.g., the speed of Gabriella as measured by the GPS can be used to deduce that she is walking), When can be provided both by the PDA or by the GPS. Viewing this from a different, more fine-grained perspective, we can imagine that one agent controlling the user profile can create a raw W4 tuple in which only the who and where are specified; another agent controlling the GPS agent create a tuple in which only where and what (i.e., the speed) are specified. Accordingly, the merging of these two raw W4 atoms into the complete one represented below can be considered as an action of the knowledge networking that produces a more complete and expressive information.

Now, let us assume that Gabriella's PDA is connected with a RFID tag reader. A specific RFID agent controls the reader and handles the event of "tag recognition" whenever a tag enters in the reading range. In this case, either the tag contains its own Who and What description in its limited memory, or the tag ID can be resolved in a database (mapping tag IDs into the associated Who-What descriptions) that the agent may access to fill in the W4 fields. Otherwise, the Who reduces to the tag ID (which enables to access to the database later) and the What is left empty. As in the previous example, the Where and When can be read from the GPS of the user. The resulting tuple is as follows:

Who: tag:#456 What: - Where: lonY, latX When: 2006/10/17:10.59am

The agent running in the knowledge network can use both the data coming from the GPS and the tag to provide a better localization of Gabriella. For example a good policy is that the RFID based location may be more accurate than the GPS one. So the resulting tuple describing Gabriella is the result of the merging between the previous ones:

Who: user:Gabriella What: walk:4km/h Where: tag#456 When: 2006/10/17:10.59am

This last example shows again a simple task of knowledge networking, in that it includes and action for relating individual atoms to increase their informative values.

4.4. W4 Knowledge Networks

Although pattern-matching techniques proved rather flexible to retrieve context information, we exploit the W4 structure to access the context repository in a semantically enriched fashion. More specifically we are working in order to link together the knowledge atoms in the knowledge space to form a network in which it would be possible to navigate from a W4 tuple to the others. Our idea is that querying such a knowledge network, instead of a flat tuple space, would lead up to an higher knowledge level resulting from the analysis, manipulation and inference upon the link structure of the knowledge network. In particular, new information could be produced by navigating the knowledge network and combining and aggregating existing information into new knowledge atoms (i.e., producing new W4 atoms in more elaborated fashion that the one exemplified in Subsection 4.3). Such new knowledge could rise from the analysis of the historical context (e.g., the location where a person spends 8 hours every day could be his workplace) or from a wide analysis over the whole W4 repository (e.g., If nobody go to work on 2007/12/25, it could be an holiday).

The realization of the knowledge network as described above is not complete, in this section we will present our preliminary works. We will expose some ways to identify both strong and weak correlation between knowledge networks, and a simple yet effective process to create new W4 atoms on the basis of the established network. In the future work section we will point out the lacks of the presented approaches and the research directions that will lead to a complete and efficient realization of the W4 knowledge network.

A relationship between knowledge atoms is caused by a relationship between the information contained in the atoms' fields. As stated in Subsection 4.2 we use pattern-matching as the basis for the query engine. In particular, for the W4 model, we can identify two types of pattern-matching correlations between knowledge atoms:

- **Same value - same field**. We can link all atoms belonging to the same user, about the same place, activity or time. Matching two or more "same value – same field" relationships, we can render complex concepts, e.g. "Everyone (every users) which is working (same activity) at the same lab (same location)".
- **Same value – different fields**. We can link atoms in which the same information compares in different fields. This kind of pattern matching can be used for augmenting the expressive level of the

information contained in the W4 atoms. For example, a knowledge atom having “When: 25/12/2006” can be linked with another atom like “Who: Christmas Day, When: 25/12/*” to add semantic information to that date.

Table 1 summarizes all possible relationships between knowledge atoms. On the principal diagonal, it is represented the “same value – same field” pattern matching. By reading the table by columns, it is possible to find all relationships between one particular atom with all other atoms in a knowledge network. For example, looking at the first column on the left, we are comparing all atoms with the same subject. The first cell is on the diagonal, so it is a “same value – same field” pattern matching. The 2nd row, 1st column cell identifies all atoms containing the different activities performed by the same subject. Then we have all atoms containing the different locations where the same subject has been, the last cell is a particular case: all atoms generated for the same user.

	Who (subject)	**What (activity)**	**Where (location)**	**When (time)**
Who (subject)	Same subject	All subjects who performed a particular activity	atom describing an indoor location (e.g. “mother’s house”)	atom describing a logical time (e.g. “xmas day”)
What (activity)	Different activity performed by the same subject	Same activity	All activities performed in the same location	All activities performed at the same time
Where (location)	All locations in which a subject has been	All location in which an activity has been performed	Same location	All locations occupied at the same time
When (time)	Same subject-different times: a living diary	All times in which an activity has been performed	All times in which a location is occupied	Same time

Table 1. Relations between the fields of W4 knowledge atoms.

This network of correlation between atoms may be used as the basis for more elaborated inference and reasoning upon knowledge network, i.e., for identifying and creating links between W4 atoms and for eventually creating new W4 knowledge atoms. To this end, we adopt an algorithmic approach which relies on a two-phase process. The first step is the identification of all possible correlations between knowledge atoms (according to Table 1), and the creation of link between W4 atoms. The second step is the generation of new knowledge atom, by analyzing which of the identified link can lead to a new W4 atom as a process of merging related atoms. Since a lot of new knowledge should be produced, and not all of the concepts should be useful, this second step should be influenced by the actual services that are accessing the knowledge network. This approach is similar to the classical data mining process, in which in a first step is devoted to identify all data sets, and then patterns and rules are inferred from sets. The underlying idea is that when analyzing vast amount of data, really a lot of relations and patterns may be discovered, but only a few of them make sense. A two step approach is general enough to let the knowledge network perform first a preliminary organization and correlation of atoms useful for all possible services that could access it, and then carry out only the new knowledge that will be reasonably used by actual services.

The following example illustrates the example of discovering new knowledge, see Fig. 3. Suppose that Gabriella’s PDA, at a certain time, creates the following tuple:

Who: user:Gabriella
What: -
Where: lonY, latX
When:
2006/12/25:12:00am

Algorithms in the knowledge network continuously analyses the network and find a lot of correlations, e.g. they find a correlation with an atom describing date */12/25, and an atom describing the spatial region in which (lonY, latX) is comprised. A new atom carrying higher level logical information may be created, such atoms states that Gabriella was in her mother's house at Christmas day.

As another example (fully elaborated in [BicMZ07]) the sensors in an environment (as represented by proper W4 atoms) can be grouped together based on the patterns of sensing (which corresponds to relating W4 atoms together based on similarities of the What field). The assumption is that sensors that sense similar data will very likely belong to the same region of the environment (e.g., sensors in the same room of a building will sense similar light patterns over time). The creation of such links among W4 sensor atoms represents the first step of the analysis process, and can abstract form any application need. Once such regions of W4 atoms are identified, new W4 data atoms can be produced representing aggregated data about each region, i.e., a single W4 tuple can be created for each region representing, in a compact way, aggregated sensorial data for that region.

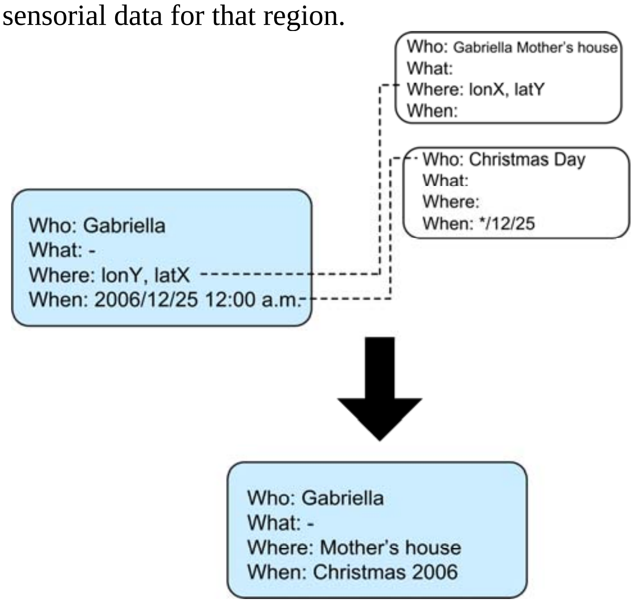

Figure 3: Creation of new atoms.

5. Conclusion and open issues

In this section we present the Testbed Prototype we have developed for putting the W4 idea at work, then we discuss the current limitation and problems of the W4 model. Follow future work, and Conclusions.

5.1. Testbed Prototype

We have already implemented a preliminary testbed prototype of a pervasive computing system for putting the W4 idea at work, and have developed some simple autonomic pervasive services within [Cas07].

The testbed prototype we have implemented is structured as follows. Users are equipped with a portable computing device (i.e., a PDAs), a localization device (i.e., a GPS), and with devices attached to

the PDA to acquire contextual information from the physical world (i.e., an RFID reader and a sensor network gateway). Of course, they are also given the possibility of connecting to the Internet via WiFi. Contextual information about the world, as gathered from RFID and sensors around, integrated with information coming from the GPS and possibly from the Web, is represented via of W4 tuples and stored by a local W4 tuple space hosted on the users' PDA. An additional remote global W4 tuple space is assumed to exist, and locally stored W4 tuples are also sent for storing in such global W4 tuples space. In the prototype, services are realized by means of application agents (i.e., autonomous software components) running locally on the user PDA and accessing, via the W4 API, both the local and Web-accessible tuple space, to retrieve the available contextual information [Fig. 4]. From the user interface viewpoint, services can interface with a local GUI client to turn W4 data into a user-centric perspective. The whole system has been realized using the Java language. The RFID reader and the sensors are accessed via JNI and sockets respectively. The Web accessible tuple space has been implemented through a Postgres database with spatial extensions, while the local tuple space is implemented by a Java Vector. User interface is provided by Google Earth for laptops and by Google Maps for PDAs.

The applications we have developed above the prototype include a sort of living diary and an interactive people map [Fig. 5]. The living diary has the goal of providing users' with real-time and historical information about their activities by *(i)* keeping track – via generation of proper W4 tuples – of all user movements and all the RFID tags and sensors it encountered around; *(ii)* enabling users to interactively query the local W4 space for historical information, and have the result visualized in the form of a tagged itinerary over GoogleMaps. The people map application considers that users in a group can decide to share their location and activity with each other. This can occur via periodically generating W4 tuples including user information, and uploading them to the global W4 tuple space. Following, it is possible for users to query the global W4 space to obtain and visualize (i.e., in GoogleMaps) real-time information about other users, and possibly additional relevant information about other activities/events around (as they can be made available via proper W4 tuples). As simple as they can be, experience with these two applications has proved us that the W4 model can make the development of pervasive services very simple.

Figure 4: User equipped with a PDA and pervasive devices can acquire contextual information from the physical world. Sensed data are integrated in W4 atoms with information coming from the Web, and stored both in local and global W4 tuple spaces.

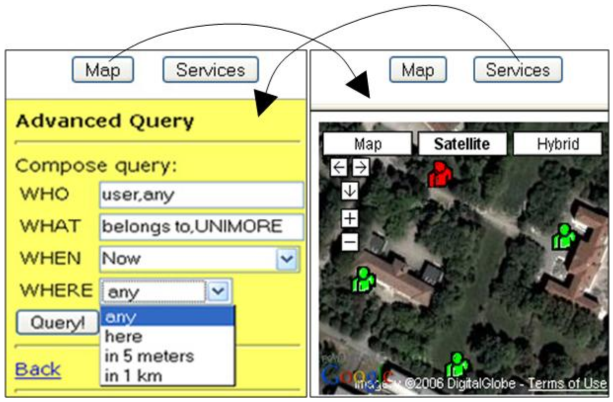

Figure 5: Map showing users real time location, and the query interface.

5.2. Discussion and current limitations

The W4 Model is our proposal for expressing contextual knowledge about the world. We presented the data model, the data access and generation, and some preliminary results about the knowledge network.

The W4 model represents the context in four fielded structures called W4 knowledge atoms. The knowledge atoms can be easily related to each other, and diverse atoms can be aggregated in a single W4 atom. The model is similar to Granular Computing principles, in that one knowledge atom can be considered as an information granule.

The proposed model tackles the challenges pinpointed in Section 2.2. The W4 data model, as presented in this paper, is general enough to represent diverse facts about the world. Its simple four-fielded structure can uniformly represents data coming from diverse sources, it can represent simple data atoms as well as aggregated atoms. The examples in Section 4 shows the expressiveness of the data model in diverse situations. The developed API to access the knowledge network layer and to inject new data is simple yet flexible in that it is based upon the classical tuple spaces mechanism, the query interface is based on expressive pattern matching upon the four fields. Respect to first works in the field of Context-awareness such as [Sch99] and [DeyAS01], the W4 Model can uniformly deal with multiple context information in a coherent way, without leading to a long list of all the characteristics of the context as [SchAW94] and [HenIR06]. Differently from the tuple based approaches ([JulR06] , [Hon02]) the W4 representation strongly structures the context representation, so that the context representation can be easily browsed. The W4 Model represents the context similarly to [Leh04], [XuC05] and [Bra06], but our approach is general purpose and able to represent a large number of context information.

About the knowledge network, we presented only some preliminary results. Our idea is to exploit the four fielded structure to identify some preferential dimensions between atoms, and create new atoms represented the inferred knowledge. In this, the proposed approach is similar to the classical data mining process, in which in a first step is devoted to identify all data sets, and then patterns and rules are inferred from sets.

Although powerful some problems and limitations affects the current W4 Model. The first criticism of the W4 approach is that it does not eradicate the problem of analyzing large amount of data, but simply passes it to a different abstraction level. It should be considered that knowledge network can take care of knowledge management duties that would have been otherwise replicated inside each service.

A serious limitation of our model is the lacks of meta data about the context, such as the freshness of the data, the source of the data, etc. that are traditionally available to the services. We plan to tackle this requirements in the future works.

The knowledge networks are in charge of continuously organize, aggregate and analyze data, this should lead to delays when services browse the network looking for some data. This is especially true if

services are looking for simple information unaffected by the knowledge network inferences and reorganization.

Another problem deals with the storage of historical information. Although historical data are useful to do inferences, for learning procedures and in general for querying, it is not possible to store all W4 atoms, some mechanism to aggregate or delete old data must be developed. Moreover it must considered that the knowledge network leads to a multiplication of new atoms coming from the inferences process.

5.3. Future Works and Research Directions

Despite the promising results achieved so far in the study of the W4 model and in its implementation, several research issues – which are to most extents general issues in context-aware and pervasive computing – still have to be faced to fully unfold its potentials.

First, we are aware that there are aspects that are not properly captured by the W4 model and that are instead, very important in general and in pervasive computing. For services (or for algorithms within the knowledge networks) to properly exploit contextual information, they may need to evaluate the accuracy and the freshness of such information, as well as their reliability. This issue, which is recognized as generally relevant to pervasive computing [RouSP07] and which we plan to deeply investigate in the near future, may require integrating factual data atoms (i.e., in our case, as represented in the four "*W*"s) with additional meta-data and make available proper algorithms and tools to manage such meta-data.

Second, the issue of self-organized knowledge networking is still to be fully unfolded, from both the algorithmic and implementation viewpoints. From the algorithmic viewpoint (as outlined in Subsection 4.4), we already have some feelings of what activities should reside in knowledge networks to enable autonomous aggregation and organization of data [BicMZ07]. However, we still have to identify a truly general-purpose algorithmic approach, suitable for a wide variety of application needs, an ambitious goal which also somewhat characterize the area of granular computing [Yao06]. From the implementation viewpoint, the issue of how to deal with decentralized and distributed data and, in particular, with enforcing distributed data analysis and networking as well as transparent data access by service, is still to be faced.

Third, we have to explore proper solutions for enabling services to dynamically inject new behaviors/algorithms within the knowledge networks. This requires identifying solutions for enabling services to specify new behavioral/algorithmic needs, and for dynamically modifying the activities inside a knowledge network without disturbing existing activities and without affecting other services. To some extents, this can be considered a specific instance of the more general issue of exploiting a shared pervasive infrastructure for serving a variety of different purposes, an issue which is assuming increasing importance in the pervasive computing and sensor networks research areas [Bau06, Ram06].

Finally, we are perfectly aware that, whenever contextual data is involved, peculiar security and privacy issues arise. On the one hand, for services to effectively perform their context-aware activities it is necessary that the contextual data they rely upon can be trusted, and that improper modifications of such data or of the algorithms that manage it within the knowledge networks should be avoided. On the other hand, it is necessary to properly control access to data to ensure that sensible and private data (e.g., user information and user activities) in the knowledge networks can be accessed and exploited only by authorized service. Although we do not have solutions at hand so far, we are confident that the large amount of research activity in security and privacy for pervasive computing [Dav03] will soon produce suitable solutions for integration in our W4 model.

5.4. Conclusions

The issue of proper organizing contextual data for their effective and meaningful exploitation by autonomic pervasive services is, and will increasingly become, a challenging issue. As the technology for producing contextual information is becoming widespread, and as a variety of diverse pervasive services

are continuously being proposed, there is need of proper solutions to represent data, pruning and organizing it into sorts of knowledge networks, making it easily available to services, possibly providing application-specific views on contextual knowledge.

In this paper, after having deeply discussed the above issues, we have shown how researchers in various areas such as context-aware computing, data mining, and granular computing, are contributing interesting solutions that can be potentially useful to tackle them. The W4 model that we have developed, as presented in this paper, proposes itself as an effective synthesis of all such work, specifically conceived for the need of future autonomic pervasive services, and paving the way for a sound and practically usable implementation of the knowledge networks concept.

Acknowledgements: Work supported by the integrated project CASCADAS (IST-027807), funded under the FET Program of the European Commission.

References

[AhuCG86] S. Ahuja, N. Carriero, D. Gelernter, “Linda and Friends”, IEEE Computer, IEEE CS Press, 19(8):26-34, 1986.

[AlCG07] G. Al-Naymat, S. Chawla, J. Gudmundsson, “Dimensionality reduction for long duration and complex spatio-temporal queries”, Symposium on Applied computing, Seoul, Korea, 2007.

[Bar03] I. Bartolini, E. Bertino, B. Catania, P. Ciaccia, M. Golfarelli, M. Patella, and S. Rizzi. “PAtterns for Next-generation DAtabase systems: preliminary results of the PANDA project”, Symposium on Advanced Database Systems, Cetraro, Italy, 2003.

[Bau06] M. Baumgarten N. Bicocchi, M. Mulvenna, F. Zambonelli , “Self-organizing Knowledge Networks for Smart World Infrastructures”, International Conference on Self-organization in Multiagent Systems, Erfurt, Germany, 2006.

[BicMZ07] N. Bicocchi, M. Mamei, F. Zambonelli, “Self-organizing Spatial Regions for Sensor Network Infrastructures”, Symposium on Pervasive and Ad-Hoc Communications, Niagara Falls, Canada, May 2007.

[BonB05] G. Bontempi, Y. Le Borgne, “An adaptive modular approach to the mining of sensor network data”, Workshop on Data Mining in Sensor Networks, Philadeplhia (PA), USA, 2005.

[Bra06] J. Bravo, R. Hervas, G. Chavira, S. Nava, “Modeling Contexts by RFID-Sensor Fusion”, Conference on Pervasive Computing and Communications Workshops, Pisa, Italy, 2006.

Bravo, J., Hervás, R., Sánchez, I., Chavira, G., Nava, S. In the special issue of Ubiquitous Computing and Ambient Intelligence. Journal of Universal Computer Science, March 2006.

[Cam03] E. Camossi, M. Bertolotto, E. Bertino, G. Guerrini, “A multigranular spatiotemporal data model”, Symposium on Advances in Geographic Information Systems, New Orleans (LA), USA, 2003.

E. Camossi, M. Bertolotto, E. Bertino: A flexible Approach to Spatio-temporal Multigranularity in an Object Data Model. International Journal of Geographical Information Science, Taylor & Francis Ltd, Volume 20, Issue 5, p. 511-534(24), 2006.

[Cas07] G. Castelli, A. Rosi, M. Mamei, F. Zambonelli, “A Simple Model and Infrastructure for Context-Aware Browsing of the World”, Conference on Pervasive Computing and Communication, New York (NY), USA, 2007.

[CegR06] A. Ceglar, J. F. Roddick, “Association Mining”, ACM Computing Surveys, ACM Press, 38(2), 2006.

[Chen04] H. Chen, F. Perich, T. Finin, A. Joshi, “SOUPA: StandardOntology for Ubiquitous and Pervasive Applications”, Conference onMobile and Ubiquitous Systems: Networking and Services”, Boston (MA), USA, 2004.

[ChoK03] C.-Y. Chong, S. Kumar, “Sensor Networks: Evolution, Opportunities, Challenges”, Proceedings of the IEEE, IEEE CS Press , 91(8): 1247-1256, 2003.

[DeGM05] I. De Fent, D. Gubiani, A. Montanari, “Granular GeoGraph: a Multi-Granular Conceptual Model for Spatial Data.”, Symposium on Advanced Database Systems, Bressanone, Italy, 2005.

[DeyAS01] A. K. Dey, G. D. Abowd, D. Salber, “A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-aware Applications”, Human-Computer Interaction, Taylor and Francis, 16(2-4):97-166, 2001.

[DeyA00] A. K. Dey, G. D. Abowd, “Towards a Better Understanding of Context and Context Awareness”, Workshop on the What, Who, Where, When and How of Context-Awareness, The Hague, The Netherlands, 2000.

[Dou04] P. Dourish, “What we talk when we talk about context”, Personal and Ubiquitous Computing, 8(1), 19-30, 2004.

[Est02] D. Estrin, D. Culler, K. Pister, G. Sukjatme, “Connecting the Physical World with Pervasive Networks”, IEEE Pervasive Computing, IEEE CS Press, 1(1): 59-69, 2002.

[Fay96] U.M.Fayyad, G. Piatetsky-Shapiro, P. Smith, “Advantages in knowledge discovery and data mining”, Data Mining to knowledge Discovery: An Overview, AAAI/MIT Press, 1996.

[GalS06] J. Galloway, S. J. Simoff, “Network Data Mining: Methods and Techniques for Discovering Deep Linkage between Attributes”, Conference on Conceptual modelling, Hobart, Australia, 2006.

[GanEH04] D. Ganesan, D. Estrin, J. Heidemann, “DIMENSIONS: Why do we need a new Data Handling architecture for Sensor Networks?”, Information Processing in Sensor Networks, Berkeley (CA), USA, 2004..

[GiP06] J. Gips, A. Pentland, “Mapping Human Networks”, Conference of Pervasive Computing and Communications, Pisa, Italy, 2006.
Eagle, N., and Pentland, A., "Reality Mining: Sensing Complex Social Systems"', J. of Personal and Ubiquitous Computing. To appear: June 2005. MOLTO SIMILE (ma cellulari …. E cambia primo autore)

[Gre01] S. Greenberg. Context as a dynamic construct. Human-Computer Interaction, 16(2-4):257-269, 2001.

[GuPZ05] T. Gu, H. K. Pung, D. Q. Zhang, “A service-oriented middleware for building context-aware services”, Journal of Network and Computer Applications, Elsevier, 28(1): 1-18, 2005.

[HenIR06] K. Henricksen , J. Indulska , A. Rakotonirainy, “Developing Context-aware Pervasive Computing Applications: Models and Approach”, Journal of Pervasive and Mobile Computing, Elsevier, 2(1):37-64, 2006.

[HigB01] J. Hightower, G. Borriello, “Location Systems for Ubiquitous Computing”, IEEE Computer, IEEE CS Press, 34(8): 57-66, 2001.

[Hon02] J. Hong., “The Context Fabric: An Infrastructure for Context-Aware Computing”, Conference on Computer Human Interaction, Minneapolis (MN), USA, 2002.

[JulR06] C. Julien , G. Roman, “EgoSpaces: Facilitating Rapid Development of Context-aware Mobile Applications”, IEEE Transactions on Software Engineering, IEEE CS Press, 32(5):281-298, 2006.

[KulD05] A. Kulakov, D. Davcev, “Data mining in wireless sensor networks based on artificial neural-networks algorithms”, Workshop on Data Mining in Sensor Networks, Newport Beach (CA), USA, 2005.

[LeeM07] D. Lee, R. Meier, “Primary Context Model and Ontology: A Combined Approach for Pervasive Transportation Services”, Workshop on Pervasive Transportation System, White Plains (NY), USA, 2007.

[Leh04] O. Lehmann, M. Bauer, C. Becker, D. Nicklas, “From home to world - supporting context-aware applications through world models”, Conference on Pervasive Computing and Communications, Orlando (FL), USA, 2004.

[ManZ06] A. Manzalini, F. Zambonelli, “Towards Autonomic and Situation-Aware Communication Services: the CASCADAS Vision”, Workshop on Distributed Intelligent Systems, Prague, Czech Republic, 2006.

[McS05] S. M. McConnell, D. B. Skillicorn, “A Distributed Approach for Prediction in Sensor Networks”, Workshop on Data Mining in Sensor Networks, Newport Beach (CA), USA, 2005.

[MotP06] G. Mottola, G. P. Picco, "Logical Neighborhoods: A Programming Abstraction for Wireless Sensor Networks", Conference on Distributed Computing in Sensor Systems, San Francisco (CA), USA, 2006.

"Lime: A Coordination Middleware Supporting Mobility of Hosts and Agents" with Amy L. Murphy and Gruia-Catalin Roman. ACM Transactions on Software Engineering and Methodology (TOSEM), vol. 15, no. 3, pp. 279-328, July 2006. MA è DIVERSO…

[NewW04] R. Newton, M. Welsh, “Region Streams: Functional Macroprogramming for Sensor Networks”, Workshop on Data Management for Sensor Networks, Toronto, Canada, 2004.

[Pat04] D. Patterson, L. Liao, K. Gajos, M. Collier, N. Livic, K. Olson, S. Wang, D. Fox, H. Kautz, “Opportunity Knocks: a System to Provide Cognitive Assistance with Transportation Services”, Conference on Ubiquitous Computing, Nottingham, United Kingdom, 2004

[Pat05] D. Patterson, D. Fox, H. Kautz, and M. Philipose. “Fine grained activity recognition by aggregating abstract object usage”, Symposium on Wearable Computers, Osaka, Japan, 2005.

T. Choudhury, M. Philipose, D. Wyatt, J. Lester. Towards Activity Databases: Using Sensors and Statistical Models to Summarize People's Lives. In IEEE Data Engineering Bulletin. March 2006.

[PauG04] E. Paulos, E. Goodman, “The familiar stranger: anxiety, comfort, and play in public places”, Conference on Human Factors in Computing Systems, Vienna, Austria, 2004.

[Ped01] W. Pedrycz, “Granular Computing: An Emerging Paradigm”, Physica Verlag, 2001.

[Pen05] A. Pentland, T. Choudhury, N. Eagle, P. Singh, "Human dynamics: computation for organizations", Pattern Recognition Letters, Elsevier, 26(4):503–511, 2005.

[Phi04] M. Philipose, K. Fishkin, M. Perkowitz, D. Patterson, D. Fox, H. Kautz, “Inferring Activities from Interactions with Objects”, IEEE Pervasive Computing, IEEE CS Press, 3(4): 50-57, 2004.

[Ram06] N. Ramanathan, L. Balzano, D. Estrin, M. Hansen, T. Harmon, J. Jay, W. Kaiser, G. S. Sukhatme, “Designing Wireless Sensor Networks as a Shared Resource for Sustainable Development”, Conference on Information and Communication Technologies and Development, Bangalore, India, 2006.

[Rou06] I. Roussaki, M. Strimpakou, C. Pils, N. Kalatzis, M.Anagnostou, “Hybrid context modeling: A location-based schemeusing Ontologies”, Workshop on Context Modeling and Reasoning, Pisa, Italy, 2006.

[RouSP07] I. Roussaki, M. Strimpakou, C. Pils, “Distributed Context Retrieval and Consistency Control in Pervasive Computing”, Journal of Networks and Systems Management, Springer Verlag, 15(1):57-74, 2007.

[SchAW94] B. Schilit, N. Adams, and R. Want. “Context-Aware Computing Applications”, Workshop on Mobile Computing Systems and Applications, Lake District, United Kingdom, 1994.

[Sch99] A. Schmidt , K. A. Aidoo , A. Takaluoma , U. Tuomela , K. Van Laerhoven , W. Van de Velde, “Advanced Interaction in Context”, Symposium on Handheld and Ubiquitous Computing, Karlsruhe, Germany, 1999.

[Ser04] J. Serrat, J. Serrano, J. Justo, R. Marín, A. Galis, K. Yang, D. Raz, Efstathios D. Sykas, “An Approach to Context Aware Services”, Network Operations and Management Symposium, Seoul, Korea, 2004.

[VerC06] F. Verhein, S. Chawla, ”Mining spatio-temporal association rules, sources, sinks, stationary regions and thoroughfares in object mobility databases”, Conference on Database Systems for Advanced Applications, Singapore, 2006.

[Wan06] R. Want, “An Introduction to RFID Technology”, IEEE Pervasive Computing, IEEE CS Press, 5(1):25-33, 2006.

[War06] J. A. Ward, P. Lukowicz, G. Tröster, T. Starner, “Activity Recognition of Assembly Tasks Using Body-Worn Microphones and Accelerometers”, IEEE Transactions on Pattern Analysis and Maching Intelligence, IEEE CS Press, 28(10): 1553-1567, 2006.

[XuC05] Chang Xu, S. C. Cheung, “Inconsistency Detection and Resolution for Context-aware Middleware Support”, Symposium on the Foundations of Software Engineering, Lisbon , Portugal, 2005.

[Yao06a] Y.Y. Yao, “Three perspectives of granular computing”, Journal of Nanchang Institute of Technology, 25(2):16-21, 2006.

[Yao06b] Y.Y. Yao, “Granular computing for data mining”, Conference on Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security, Kissimmee (FL), USA, 2006.